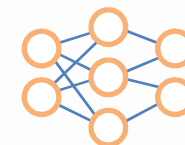


***LINKgroup***

Distance Learning System



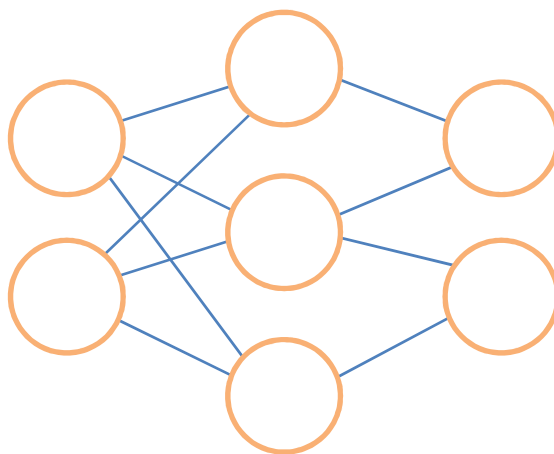
# Neuronska mreža

Python Data Access and Processing

# Veštačka neuronska mreža

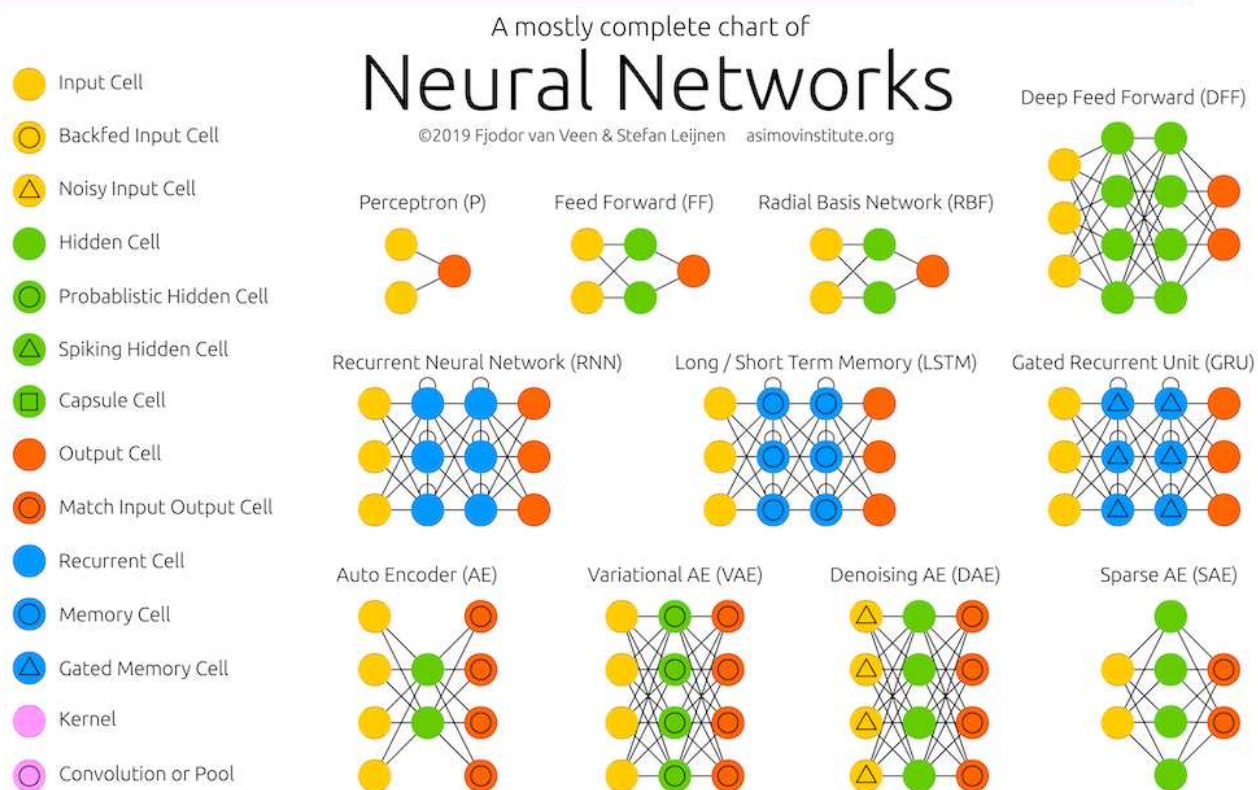
---

- Veštačka neuronska mreža je skup uzajamno povezanih čvorova koji imaju različite, najčešće numerički predstavljene, karakteristike
- Veštačka neuronska mreža se može realizovati u različitim oblicima (Feed Forward, Radial, Reccurent, Convolutional, Modular...)



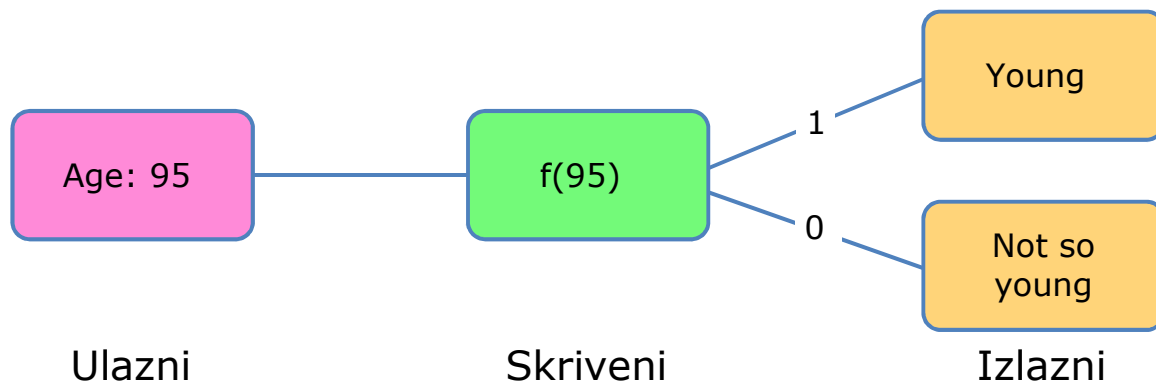
# Tipovi neuronskih mreža

<https://www.asimovinstitute.org/author/fjodorvanveen/>



# Struktura veštačke mreže

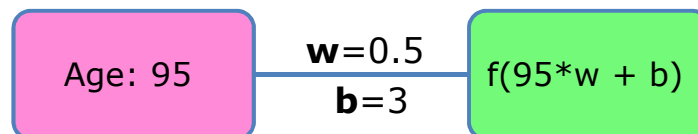
- Najmanji sastojak neuronske mreže jeste jedan čvor (neuron)
- Čvor može imati različitu strukturu u zavisnosti od svoje pozicije u mreži
- Pozicije u mreži mogu biti: **ulazne**, **skrивene** i **izlazne**
- Mreža ima jedan sloj ulaznih čvorova i jedan sloj izlaznih čvorova, dok može biti više slojeva sa skrivenim čvorovima



# Weight i bias

---

- Skriveni čvorovi imaju ulazne vrednosti (dobijaju ih od ulaznih čvorova), težinu (weight) ulazne vrednosti i sklonost (bias)



- Skriveni čvorovi imaju ulazne vrednosti (dobijaju ih od ulaznih čvorova), težinu (weight) ulazne vrednosti i sklonost (bias)
- **Weight** je koeficijent aktivacije noda
- **Bias** je offset kojim se pojačava intenzitet izlaza noda
- Ako je ulazna vrednost 95, aktivacija će biti:  $f(95 * 0.5 + 3) = f(50.5)$

# Aktivaciona funkcija

---

- Logika po kojoj odlučujemo da li će izlazna vrednost čvora biti **da** ili **ne**

- Sigmoid

$$1 / (1 + \exp(-x))$$

- ReLU

- Threshold

- Tanh

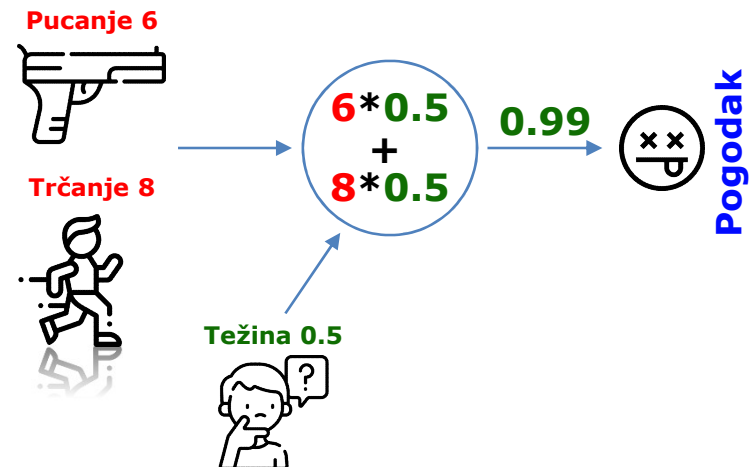
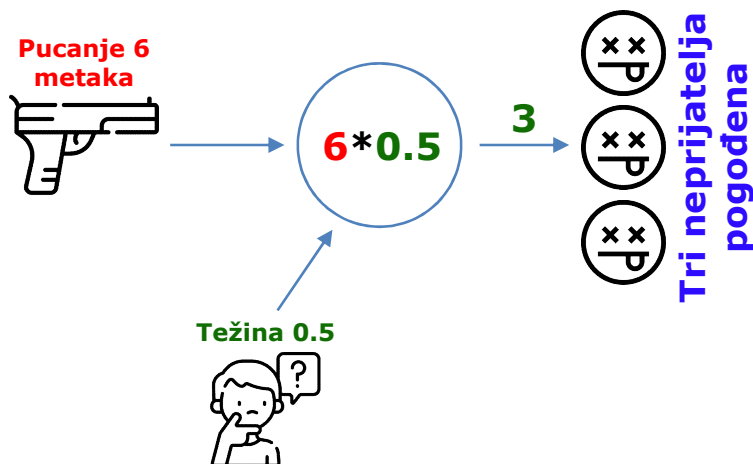
- Softmax

$$\exp(x) / \sum(\exp(x))$$

- ...

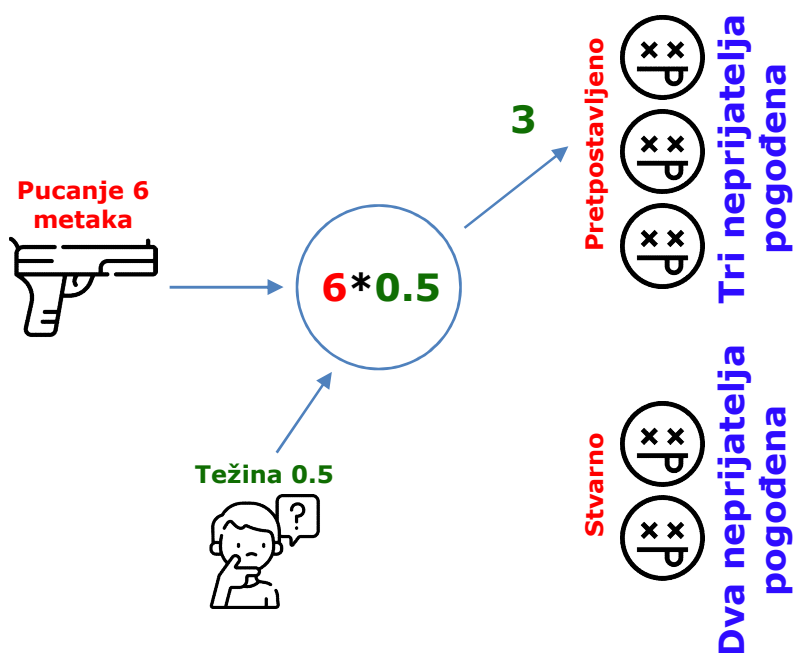
# Forward propagacija

- Pretpostavlja se izlazna vrednost ( $y^{\wedge}$ ) na osnovu ulaznih vrednosti i "težine"



# Greška (loss / cost funkcija)

- Greška je razlika između predviđenog i stvarnog podatka



Stvarno

$$(2 - 3)^2$$

Pretpostavljeno

Greška je: **1**

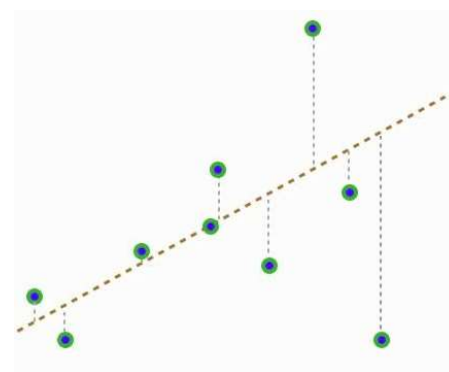


# Različite funkcije za pronalaženje greške

- Mean Square
- Mean Absolute
- Huber

```
def loss_f(pred, target):  
    return (  
        (pred-target)**2  
    ).mean()
```

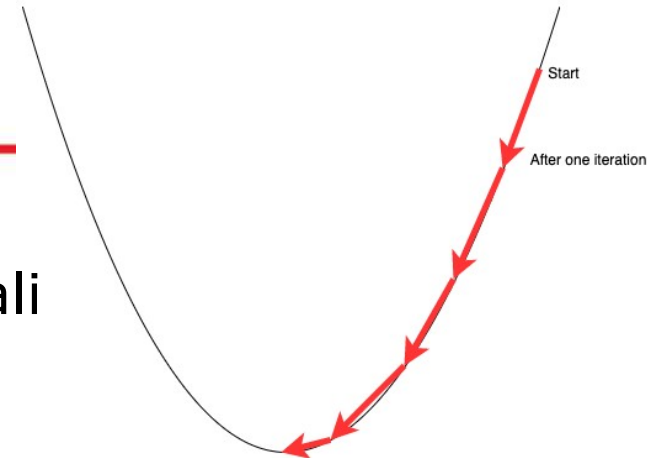
- Hinge Loss
- Cross Entropy
- Multi class Cross Entropy



# Optimizacija

---

- Optimizacija je procedura u kojoj modifikujemo parametre kako bi povećali mogućnost pogotka
- Algoritmi optimizacije
- Gradient descent
  - Stochastic
  - Batch
  - Min batch
  - ...



```
def opt(x,y,pred):  
    return np.dot(  
        2*x , pred-y  
    ).mean()
```

# Back propagation

---

- Parametri (težine i bias-i) se modifikuju u skladu sa preporukom optimizatora

$$w = w - grad * 0.05$$

# Credits

---



<https://www.flaticon.com/authors/freepik>



<https://www.flaticon.com/authors/nikita-golubev>

[https://dev.to/\\_akshaym/neural-network-basics-gradient-descent-4cej](https://dev.to/_akshaym/neural-network-basics-gradient-descent-4cej)

<https://mubaris.com/posts/linear-regression/>

**LINKgroup**